Neural Networks 51 (2014) 17–25

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Neural network for solving convex quadratic bilevel programming problems

Xing He [a], Chuandong Li [a,*], Tingwen Huang [b], Chaojie Li [c]

[a] School of Electronics and Information Engineering, Southwest University, Chongqing 400715, PR China
[b] Texas A & M University at Qatar, Doha 5825, Qatar
[c] School of Science, Information Technology and Engineering, University of Ballarat, Mt Helen, VIC 3350, Australia

ARTICLE INFO

Article history:
Received 2 August 2013
Received in revised form 14 October 2013
Accepted 18 November 2013

Keywords:
Neural network
Convex quadratic bilevel programming problems
Nonautonomous differential inclusions
Nonsmooth analysis

ABSTRACT

In this paper, using the idea of successive approximation, we propose a neural network to solve convex quadratic bilevel programming problems (CQBPPs), which is modeled by a nonautonomous differential inclusion. Different from the existing neural network for CQBPP, the model has the least number of state variables and simple structure. Based on the theory of nonsmooth analysis, differential inclusions and Lyapunov-like method, the limit equilibrium points sequence of the proposed neural networks can approximately converge to an optimal solution of CQBPP under certain conditions. Finally, simulation results on two numerical examples and the portfolio selection problem show the effectiveness and performance of the proposed neural network.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Bilevel programming problems (BPPs) are hierarchical optimization problems in which the constraint region is implicitly determined by another optimization problem. The BPP can be formulated as follows:

$$\begin{aligned} &\min_x \; F(x, y) \\ &s.t. \;\; H(x, y) \le 0 \\ &y \in \begin{cases} \min_y \; f(x, y) \\ s.t. \;\; h(x, y) \le 0 \\ x \in X, \quad y \in Y \end{cases} \end{aligned} \tag{1}$$

where $X \subset R^n$, $Y \subset R^m$, and H, h are vector valued functions of dimensions p and q. F and f are real-valued functions of appropriate dimensions.

Numerous applications in science and engineering, such as network design, transport system planning, management and economic policy, can be formulated as BPP. Fernandez-Blanco, Arroyo, and Alguacil (2012) constructed a general bilevel programming framework for alternative market-clearing procedures dependent on market-clearing prices. Using bilevel programming and swarm intelligence technique, Zhang, Zhang, Gao, and Lu (2011) presented a competitive strategic bidding optimization problem in electricity markets. Yang, Zhang, He, and Yang (2009) constructed a bilevel programming model for the flow interception problem with customer choice. So many researchers have made deep research in this field, including the theory, algorithm and application of bilevel programming (Amouzegar, 1999; Dempe, 2002; Etoa, 2010; Luo, Pang, & Ralph, 1996; Teng & Li, 2002; Vicente, Savard, & Júdice, 1994; Wang, Jiao, & Li, 2005). In the past years, a variety of numerical algorithms have been developed for BPP. However, in many engineering applications, real-time solutions are often needed. For such real-time applications, neural networks based on circuit implementation (Hopfield & Tank, 1985) are more competent.

Over the years, neural networks for optimization and their engineering applications have been widely investigated. Tank and Hopfield applied the Hopfield network for solving linear programming problems (Hopfield & Tank, 1985; Tank & Hopfield, 1986), which motivated the development of neural networks for solving linear programming (Liu, Cao, & Chen, 2010; Wang, 1993; Xia, 1996; Xia & Wang, 1995), variational inequalities (Cheng, Hou, & Tan, 2008; Gao, Liao, & Qi, 2005; Hu & Wang, 2006, 2007), nonlinear programming (Bian & Chen, 2012; Forti, Nistri, & Quincampoix, 2004, 2006; Hosseini, Wang, & Mohamma, 2013; Liu, Dang, & Huang, 2013; Liu, Guo, & Wang, 2012; Liu & Wang, 2011, 2013; Xia & Wang, 2004) and so on. These neural networks are essentially governed by a set of dynamic systems characterized by an energy function, which is the combination of the objection function and constraints of the original optimization problem, and

* Corresponding author. Tel.: +86 23 65103199.
E-mail addresses: hexingdoc@hotmail.com (X. He), licd@cqu.edu.cn, cd_licqu@163.com (C. Li), tingwen.huang@qatar.tamu.edu (T. Huang), cjlee.cqu@163.com (C. Li).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.11.015

three common techniques, such as penalty functions, Lagrange functions and primal and dual functions, which are used to construct neural networks for solving the optimization problem.

Recently, neural networks for solving BPP have received attention in the literature (Shih, Wen, Lee, Lan, & Hsiao, 2004), (Hu, Gao, Fu, & Lv, 2010; Lan, Wen, Shih, & Lee, 2007; Lv, Chen, & Wan, 2010; Lv, Hu, Wang, & Wan, 2008; Sheng, Lv, & Xu, 1996). Based on the Frank–Wolfe method, Sheng et al. (1996) first proposed a neural network to solve a class of BPPs appearing in the algorithm. Shih et al. (2004) utilized the dynamic behavior of neural networks to solve multiobjective programming and multilevel programming problems. Lv and his colleagues (Hu et al., 2010; Lan et al., 2007; Lv et al., 2010, 2008) presented neural networks for solving the bilevel linear programming problem (BLPP), convex quadratic BPP and nonlinear BPP. These neural networks for solving BPP are based on Lagrange functions. However, due to the use of Lagrange multipliers, the number of state variables increased doubly, which enlarged the scale of network. Recently, some neural networks (Liu et al., 2010, 2013, 2012; Liu & Wang, 2011) for nonlinear optimization are constructed based on the penalty function, which can be used to reduce the scale of neural networks. Therefore, there is an urgent and significant need to reduce the scale of neural networks for solving BPP.

In this paper, following the Karush–Kuhn–Tacker optimality conditions (Facchinei, Jiang, & Qi, 1999), we first transform the convex quadratic bilevel programming problems (CQBPPs) into a single level problem. Then an approximation equivalent nonlinear optimization problem can be obtained through smoothing the single level problem. In order to solve the approximation equivalent problem effectively, based on the method of penalty functions and the theory of differential inclusions, nonautonomous neural network and neural sub-networks can be constructed. Compared with existing neural networks for CQBPP (Lv et al., 2010), the true power and advantage of our neural networks lie in simple structure and the least number of state variables, and their dynamical behavior and optimization capabilities are analyzed in the framework of nonsmooth analysis (Clarke, 1983) and the theory of differential inclusions (Aubin & Cellina, 1984). It is shown that the limit equilibrium points sequence of the proposed neural networks can approximately converge to an optimal solution of CQBPP under certain conditions. Simulation results on numerical examples and the portfolio selection problem show the effectiveness and performance of the neural network for solving CQBPP.

The remainder of this paper is organized as follows. In the next section, the preliminaries relevant to CQBPP are introduced. In Section 3, the nonautonomous neural network is derived. The convergence of the proposed neural network is proved in Section 4. In Section 5, neural sub-networks for solving CQBPP are constructed. Simulation results on two numerical examples and the portfolio selection problem are given in Section 6 to demonstrate the effectiveness and performance of the neural network. Finally, Section 7 concludes this paper.

Notation: Given column vectors $x = (x_1, x_2, \dots, x_n)^T$ and $y = (y_1, y_2, \dots, y_n)^T$, $\langle x, y\rangle = x^T y = \sum_{i=1}^{n} x_i y_i$ is the scalar product of x, y, and $\|x\|_1 = \sum_{i=1}^{n} |x_i|$, $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$, $x^2 = [x_1^2, x_2^2, \dots, x_n^2]^T$, $\sqrt{x} = [\sqrt{x_1}, \sqrt{x_2}, \dots, \sqrt{x_n}]^T$, $1_n = \underbrace{[1, 1, \dots, 1]}_{n}^{T}$, $R_+^1 = [0, +\infty)$, $\dot{\varepsilon}(t)_{2m+n+q} = \underbrace{[\dot{\varepsilon}(t), \dot{\varepsilon}(t), \dots, \dot{\varepsilon}(t)]}_{2m+n+q}^{T}$.

2. Preliminaries

In this section, some models, assumptions and lemmas about CQBPP are introduced, which are needed in the following development. If F and f are quadratic functions, and H and h are linear constraints, problem (1) gives rise to the following

$$\begin{aligned} (UP)\ \min_{x\geq 0}\ & F(x,y) = \frac{1}{2}(x^T, y^T)\begin{pmatrix} C_1 & C_3 \\ C_3^T & C_2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + c_1^T x + d_1^T y \\ & s.t.\quad A_1 x + B_1 y \leq b_1 \\ (LP)\ \min_{y\geq 0}\ & f(x,y) = \frac{1}{2} y^T Q y + y^T D x + d_2^T y \\ & s.t.\ A_2 x + B_2 y \leq b_2 \end{aligned} \tag{2}$$

where $c_1 \in R^n, d_1, d_2 \in R^m, C_1 \in R^{n\times n}, Q, C_2 \in R^{m\times m}, D, C_3^T \in R^{m\times n}$, $A_1 \in R^{p\times n}, B_1 \subset R^{p\times m}, A_2 \in R^{q\times n}, B_2 \subset R^{q\times m}, b_1 \in R^p, b_2 \in R^q$. The term (UP) is called the upper level problem and (LP) is called the lower level problem. At the UP, the decision maker has to choose first a vector $x \in X$ to minimize his objective function F; then under this decision the LP decision maker has to select the decision vector $y \in Y$ that minimizes his own objective f. Throughout the rest of the paper, we make the following assumptions.

Assumption 1. The constraint region of the above bilevel programming problem

$$S = \{(x, y) : x \geq 0, y \geq 0, A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2\}$$

is nonempty and compact.

Assumption 2. $C = \begin{pmatrix} C_1 & C_3 \\ C_3^T & C_2 \end{pmatrix}$ and Q are positive semi-definite matrices.

From Facchinei et al. (1999), we can reduce the bilevel programming problem to the one-level programming problem by replacing the lower-level problem with its Karush–Kuhn–Tacker (KKT) optimality condition. The KKT reformulation of CQBPP follows:

$$\begin{aligned} & \min F(x,y) \\ & s.t.\quad A_1 x + B_1 y \leq b_1 \\ & A_2 x + B_2 y \leq b_2 \\ & Qy + Dx + d_2 + B_2^T u - \upsilon = 0 \\ & u^T (b_2 - A_2 x - B_2 y) = 0 \\ & \upsilon^T y = 0 \\ & x, y, u, \upsilon \geq 0 \end{aligned} \tag{3}$$

where $u \in R^q$, $\upsilon \in R^m$. Problem (3) is non-convex and non-differentiable, moreover the regularity assumptions which are needed for successfully handling smooth optimization problems are never satisfied and it is not good for using the neural network approach to solve the problem. Dempe (2002) presented the smoothing method for solving BPP. Following this smoothing method, we can propose a neural network approach to solve problem (3) in the next section.

Let $\varepsilon \in R_+$ be a parameter. Define the function $\Phi_\varepsilon : R^2 \to R$ by

$$\Phi_\varepsilon(a, b) = \sqrt{a^2 + b^2 + 2\varepsilon} - a - b.$$

The important property of this function can be stated in the following result.

Lemma 1. *For every $\varepsilon > 0$, we have*

$$\Phi_\varepsilon(a, b) = 0 \Leftrightarrow a > 0,\ b > 0,\quad ab = \varepsilon$$

Download English Version:

https://daneshyari.com/en/article/404003

Download Persian Version:

https://daneshyari.com/article/404003

Daneshyari.com